JOINT INTEROPERABILITY & ENGINEERING ORGANIZATION

CENTER FOR SOFTWARE

Management Plan MP

30 April 1995

SOFTWARE INSTALLATION PLAN (SIP)

FOR THE

AIRFIELDS SYSTEM

Version 2.0.1
CM Number: LL-521-05-05

(D R A F T)

Revised 16 February 1996

SUBMITTED BY:

JAMES MOODY
Chief, General
Applications Division

APPROVED BY:

SAMUEL PUCCIARELLI
Chief, Software
Development Department

Copies of this document may be obtained from:

The Director CFSW
Attn: Code JEXAG
5600 Columbia Pike
Falls Church, VA 22041

ACKNOWLEDGEMENT

This document was prepared for the Defense Information Systems Agency (DISA), Joint Interoperability and Engineering Organization (JIEO), Center for Software (JEX), Software Development Department (JEXA), General Applications Division (JEXAG).

This Software Installation Plan (SIP) provides a specific detailed plan for installing the Airfields System software at the user site. Included in this document are plans for getting started/preparations, user training, and conversion from existing systems. This version of the system which has been re-engineered/re-hosted is intended to demonstrate the practicality of using Ada 95 as a programming language in support of the conversion and re-engineering of the Worldwide Military Command and Control System (WWMCCS) applications to the Global Command and Control System (GCCS) environment.

Any questions, comments, or considerations relative to this Software Installation Plan should be directed to the following:

Global Command and Control System (GCCS) Hotline
DSN: 653-8681
Commercial: (703) 735-8681

CONTENTS

SECTION 1. SCOPE

1.1 Identification . The Airfields system provides the Worldwide Military Command and Control System (WWMCCS) community with a wide range of data about free world airfields. All data is supplied by the Defense Mapping Agency Aerospace Center (DMAAC) and is updated monthly. The Airfields Retrieval system has been identified as a Global Command and Control Migration System and was re-engineered from COBOL to the Ada 95 language. It provides the capability to print the One-Line, One-Page Summary, Multi-Page, Selective Data Retrieval (not part of Version 1.0.4 delivery), and Turnaround reports both on- and off-line.

1.2 System Overview . The functional proponent for the Airfields system is the Joint Staff's Logistics Directorate (J4). The office of primary responsibility (OPR) is the Operations Planning Division under the Joint Interoperability and Engineering Organization (JIEO). The designated development Agency (DDA) is the Center for Software (JEX), Software Development Department (JEXA), General Applications Division (JEXAG).

The original system was hosted on proprietary hardware and is written mostly in COBOL. A small percentage of the software is written in FORTRAN.

The Airfields System has been in existence for approximately twenty years. In the mid to late 1980's, the Defense Mapping Agency Aerospace Center (DMAAC) changed the database format which resulted in the need to do a total redesign of the WWMCCS version of the system from COBOL 68 to COBOL 74. During that period, the access method also changed from Honeywell Indexed Sequential Processing (ISP) files to a flat file format.

Historically, WWMCCS users access the system approximately 100 times per month. The database is owned by the Defense Mapping Agency Aerospace Center (DMAAC) and contains data on approximately 44,000 airfields and consists of over one million records.

The Airfields database is a flat file database that is currently resident on the Worldwide Military Command and Control Systems (WWMCCS) Honeywell mainframe. Reverse engineering was used to re-host the database using the Relational Database Management System (RDBMS) in the Oracle Standard Query Language (SQL). The system runs under a Sun Solaris 2.3 environment. A commercial-off-the-shelf (COTS) Graphical User Interface (GUI), Screen Machine, is utilized at the front end of the system.

The system complies with GCCS Integration Standards and employs many standards such as the windowing capability and an extensive Help facility to aid the user with system operation. The primary

operational sites include the Worldwide Military Command and Control System (WWMCCS) community and the Joint Staff.

1.3 Document Overview . The purpose of this Software Installation Plan (SIP) is to provide the information needed to install the Airfields System into the Global Command and Control System's (GCCS') Common Operating Environment. Installers should be aware that Airfields data is classified Secret/No Foreign Dissemination and the utmost precautions should be taken when handling the data. Installers should possess a SECRET (or higher) security clearance. Should questions arise concerning the handling of the data being installed, please contact the GCCS Automated Data Processing (ADP) Security Officer.

1.4 Relationship to Other Plans . This Software Installation Plan (SIP) bares no relationship to other project management plans.

2. REFERENCED DOCUMENTS

a. Department of Defense, Military Standard Software Development and Documentation , MIL-STD-498, 5 Dec 1994

b. Data Item Description (DID) number DI-IPSC-81428, Software Installation Plan (SIP) , 5 Dec 1994

c. Joint Interoperability & Engineering Organization (JIEO), Washington, DC, Software Development Plan (SDP) , (Draft) 20 January 1995

d. Joint Interoperability & Engineering Organization (JIEO), Washington, DC, Software Requirements Specification (SRS) (Draft), 20 January 1995

3. **INSTALLATION OVERVIEW**

3.1 Description . The Airfields system will be installed on a SunSparc workstation in the GCCS Common Operating Environment (COE). Airfields runs under Solaris 2.3 and database manipulations are handled via the Oracle Standard Query Language (SQL). An Open Database Connectivity (ODBC) package is utilized to tie the Airfields application and the Airfields database together. A commercial-off-the-shelf (COTS) Graphical User Interface (GUI) package called Screen machine is utilized at the front end of the system. In order to operate fully, the software listed in Section 4.1.2 - Software Inventory, must be resident on the host system. Specifics concerning Airfields availability, access method(s), sites to receive (or network into) the software, and scheduled dates and method of installation will be addressed by the GCCS System Administrator.

3.2 Contact Point . The following is the point of contact should there be questions regarding the installation of the system:

Center for Software
Software Development Department
Conventional Command and Control Division
5600 Columbia Pike
Falls Church, VA 22041

Commercial Phone Number: (703) 681-2599

3.3 Support Materials . Support materials needed for installation include all software listed in Section 4.1.2. Approximately one (1) gigabit of storage space is required to load and run the Airfields application. All software is being provided on eight millimeter tape(s) by the General Applications Division (JEXAG) of the Software Development Department (JEXA) located at 5600 Columbia Pike in Falls Church, Virginia. All data is supplied (and will be updated monthly) by the Defense Mapping Agency Aerospace Center (DMAAC). Continuous feed 132 character wide printer paper is required to print most reports. Some printers may require 8.5 by 11 bond paper.

Airfields database load procedures are provided and a copy is attached to this document as Appendix A.

3.4 Training . User training may be provided by the GCCS System Administrator. Training dates have not been identified.

3.5 Tasks . Tasks associated with the installation of the Airfields system, the Airfields database, and all support materials and software include the following:

a. Overall planning, coordination, and preparation for installation will be provided by the Ada 95 Early Adoption Project team of the Software Development Department.

b. Providing personnel for the installation team will be accomplished by GCCS.

c. All manuals applicable to the installation will be provided by the Ada 95 Early Adoption Project's Configuration Manager/Documentation specialist.

d. Training and other planning will be provided by the GCCS System Administrator.

e. Computer support and technical assistance for the installation will be provided by the GCCS and the Ada 95 Early Adoption Project of the Software Development Department.

f. Conversion from the current system is being accomplished by the Ada 95 Early Adoption Project of the Software Development Department.

3.6 Personnel. Installation will be accomplished by GCCS personnel with segmentation development being accomplished by the Ada 95 Early Adoption Project team members. All installation personnel are Computer Specialist and Computer Scientists who possess a myriad of automated data processing skills.

3.7 Security and Privacy. The database is classified Secret/No Foreign Dissemination (SNF). On-line reports from the retrieval system are marked with the highest classification of data actually reported. Continental U.S.A. (CONUS) data is unclassified. All other airfields can be classified up to SNF. Users are advised to control classified reports properly. Should questions arise concerning ADP security, contact the GCCS system security office.

4. **SITE-SPECIFIC INFORMATION FOR SOFTWARE CENTER OPERATIONS STAFF**.

The following paragraphs identify the computer center(s) and centralized/networked software installations needed for users to access the Airfields system.

4.1 GCCS Community . The software will be installed or made available (through a Wide Area Network) to the entire GCCS Community.

4.1.1 Schedule . For a schedule of tasks to be accomplished during installation, contact the GCCS System Administrator.

4.1.2 Software Inventory . The following represents the software needed to support the installation of the Airfields segment of the GCCS:

Oracle Version 7.1	-	Unclassified
Open Database Connectivity (ODBC) Version 2.0	-	Unclassified
Solaris 2.3	-	Unclassified
Airfields Executables Version 2.0	-	Unclassified
Oracle database Creation Scripts	-	Unclassified
Airfields Database Tables Version 2.0	-	Secret/NOFORN

Open Database Connectivity Licensing will be managed by the GCCS.

4.1.3 Facilities . The physical facility(ies) and accommodations needed during the installation period will be determined by GCCS. Hardware that must be available includes a SunSparc 1000 workstation and printers for report production.

4.1.4 Installation Team . The installation team has been determined by the GCCS. Associated tasks follow:

a. Loading the system executable - GCCS personnel

b. Database Load - Airfields Database Team (3)

c. Segmentation Development - Computer Scientists (2)

d. Port ODBC to Solaris 2.3 - Computer Specialists (3)

e. Acceptance testing - Computer Scientists (2)

f. Provide full documentation - Computer Specialist (1)

4.1.5 Installation Procedures . The loading of the system will be accomplished by GCCS using an eight millimeter tape provided by the Software Development Department. The tape will contain an executable(s) of the Airfields software. The database load will be accomplished using Airfields data supplied by the Defense Mapping Agency Aerospace Center (DMAAC)and Oracle scripts and data managers to accomplish the database load. Oracle scripts and data managers are provided by the Ada 95 Database Team. The data is classified Secret NOFORN, therefore caution must be taken when installing, accessing, and printing the data. Users who access the system are required to possess at least a Secret security clearance and should have the need-to-know to access the information. Any questions concerning security should be directed to the GCCS ADP Security Officer.

The data management portion of the system will be modified to use Open Database Connectivity (ODBC) commands (rather than embedded Standard Query Language (SQL). This accomplishes what is more commonly know as "binding" the application to the database. For information regarding ODBC, refer to Microsoft's ODBC Version 2.0 Programmer's Reference and SDK Guide. Additional information regarding ODBC can be found in the Readme file located at /h/Airfieles/license. On "Danny", the development system, the Readme file is located at user1/airfields/lpatton/segmentation/Airfields/license

Acceptance testing will be accomplished and a log of the results documented. Parallel runs will be done to ensure consistency between the old system and the new.

4.1.6 Data Update Procedures . The Defense Mapping Agency Aerospace Center (DMAAC) provides an update to the Airfields database monthly. Using the update tapes supplied by DMAAC, the Airfields database will be updated monthly using update software developed by the Software Development Department.

Instructions for accomplishing the initial database load can be found in Appendix A of this Software Installation Plan (SIP).

4.1.7 Segementation Procedures . For detailed instructions on accomplishing segementation, refer to the GCCS System Integration Standards document, Version 1.0, dated 26 October 1994.

4.1.8 Segmentation Install and Deinstall . For detailed

instructions on GCCS tools required for GCCS segmentation install and deinstall, refer to the following GCCS documentation:

(a) GCCS Implementation Procedures for AIC
GCCS Version 2.1, Route 0, Final
Dated 27 September 1995
CM # LL-500-103-18

(b) GCCS System Administration Manual
Route 0, Final
Dated 29 September 1995
CM # LL-500-29-10

5. **NOTES**

5.1 Terms and Abbreviations

ADP	Automated Data Processing
CFSW	Center for Software
COE	Common Operating Environment
DDA	Designated Development Agency
DID	Data Item Description
DISA	Defense Information Systems Agency
DMAAC	Defense Mapping Agency Aerospace Center
GCCS	Global Command and Control Systems
GUI	Graphical User Interface
JIEO	Joint Interoperability & Engineering Organization
OPR	Office of Primary Responsibility
RDBMS	Relational Database Management System
SDP	Software Development Plan
SNF	Secret/No Foreign [dissemination]
STD	Standard
SQL	Standard Query Language
SRS	Software Requirements Specification
WWMCCS	Worldwide Military Command and Control Systems

APPENDIX A

AIRFIELDS DATABASE LOAD PROCEDURES

Airfields data is owned by the Defense Mapping Agency Aerospace Center (DMAAC). There are approximately 44,000 airfields in the database which consists of over one million records. DMAAC has provided the Defense Information Systems Agency (DISA) with an initial load tape that contains all 44,000 airfields, all the data necessary to populate the Global Command and Control System (GCCS) Airfields database. Updates to the database will also be provided by DMAAC. DISA will be responsible for updating the GCCS Airfields database. The following paragraphs explain how the loading of the database is accomplished.
The following standards are used throughout this document:

STANDARD	EXPLANATION
Bold	Computer supplied instructions, responses, prompts.
Italicized	Exact key strokes, file name, table name
Bracketed information [ ]	User must supply the directory structure
Bracketed & Italicized [*ab*]	important notes, explanation to user about db load.
Information enclosed in < >	Identifies a specific keystroke on the keyboard

Assumption: The user understands how to log into the UNIX environment. There are no other assumptions made in these instructions. This document was written for the user with little UNIX experience. It has been written as simple as possible to accommodate the user with no UNIX experience, however we recommend that some UNIX is known before attempting an Airfields database load. (At least have someone UNIX literate available for any UNIX related questions.)

UNIX commands (mentioned in this document, helpful for completion of load):
pwd (print working directory)
command at the Unix Shell level (%) will show the current

directory location.

cd (change directory)changes to the specified directory.
(Ex: cd /home/scripts/step01 - changes from the current directory to the specified directory; in this case /home/scripts/step01.

ls Lists the files in the current directory.

(ls -l Lists the files, the owner and date/time information)

For the use of other Unix commands or a more detailed explanation of the aforementioned commands type:
man *(space)* command at the Unix prompt.
Manual pages with instructions on the specified command will display.

Ex:
[This is an excerpt from a man cd done in the UNIX environment on a SUN named danny]

danny% man cd
Reformatting page. Wait... done
NAME
cd -change working directory

SYNOPSIS
cd [directory]

DESCRIPTION
If directory is not specified, the value of shell parameter $HOME is used as the new working directory. If directory specifies a complete path starting with ` / ', ` .', or ` .. ', directory becomes the new working directory. If neither case applies, cd tries to find the designated directory relative to one of the paths specified by the $CDPATH shell variable. $CDPATH has the same syntax as, and similar semantics to, the $PATH shell variable. cd must have execute (search) permission in directory.

Because a new process is created to execute each command, cd would be ineffective if it were written as a normal command; therefore, it is recognized by and is internal to the shell.

Airfields Database Load (ADL) Instructions.

Step00

Log onto the system. If in the Open Windows environment, it is helpful to have more than one window open (at least three).

1.
SQL Plus running (identifiable by the “SQL>” prompt)) at all times

2 .
Unix Shell (identifiable by the percent sign (%) prompt) at all times

3.
Working window - used to move to different working environments as instructed.

I. Setup the Airfields database load environment.
A. Read the scripts from tape.
The airfields scripts tape was created by using tar(create tape archives.)
To read the tape, type the following at the Unix prompt.
%**tar -xvf** *tape_drive_name*

Once the tape is read, the following directory structure will be in place:
airf_db_load

step01 step02 step03 step04...step10 cleanup

The directories step01, step02, step03, step04...step10 correspond to the 10 steps necessary to complete an ADL.

The data from DMA’s ADM comes on a tape is in “dd” format. The following script will read 10 files from a tape.

```
#!/bin/sh

for I in 01 02 03 04 05 06 07 08 09 10

do
echo$I
dd  if = /dev/rmt/1mn > directory_structure/file$I*
echo done$I
done
```

where /dev/rmt/1mn => tape drive name, and the file will be located in the directory structure specified with the name “file” concatenated with the value of the variable I.

*The standard file names used have been file$I for regular data,

and rmks$I for remarks data, because we received remarks on a separate tape.

DMA usually sends a hardcopy of the first file on the tape (notes) which contains
a. List of their tables
b. The order in which the above tables data is stored on the tape
c. The number of rows in each table.

From this hard copy we obtain the maximum value of "I", which is the number of data files on the tape. If not enclosed, read the first file from the tape. This is an electronic version of the hard copy.

All data except remarks from DMA should be read into the step02 directory. Remarks data should reside in the step07 directory (This is done from the Unix Shell window). Upon completion of the ADL the data may be moved to a separate data directory for configuration management purposes, however it is necessary that the data reside in the directory with the scripts in steps 02 and 07 while loading.

Step01

Scripts executed in this step: *a1_create_dma_tables.sql*
Approximate Time*:
1hr.

Description *:*

a1_create_dma_tables.sql script executes a script for each table in the ADM database which performs a create for that particular table. As each script executes, the initial step performs a table drop. This is a precautionary step to ensure that a table, which may contain garbage or data otherwise incorrect or corrupt, is not utilized. Next, the table is created and data definitions of the table (element name, element type, element size, and key identification) are created. The naming convention of each script contained within *a1_create_dma_tables.sql* is as follows: the words '*create_dma*' concatenated with *each_table_name* and *.sql* extensions at the end. As the above file executes, Oracle will respond with ***table dropped***, or ***table created*** when appropriate. This is accomplished on a table by table basis until all tables identified in the scripts have been created.

How to:

Change to the step01 directory. Start Sqlplus.
[Use (cd ..) to go backward one directory,
(cd directory_name) to go to a directory directly below the current

directory].

At the directory prompt type:
%sqlplus username/password

At the sql prompt (SQL>) execute the first step of the ADL by typing the following.
SQL>@***a1_create_dma_tables.sql*** *<RETURN>/<EXECUTE>*

[Note: Step06 can be done now. Originally, we obtained the remarks data late so they were done seperately.]

Exit from SQL.

Step02
Scripts executed in this step: *a2_load_dma_tables*
Approximate Time*:
1.5 hr.

a2_load_dma_tables.sql loads the data read in step00 into the tables created in step01. The *a2_load_dma_tables* file calls sqlloader specifying the control (*.ctl*) script and input file (input_file##.dat) for each table that has been identified in Step01. above. Sqlloader then loads the data from the *.dat* file into the table specified in the ctl file. Data in the ctl file is vertical line (|) delimited. The naming convention of the ctl files contained in the *a2_load_dma_tables* file is *'create_dma'* at the beginning and *.ctl* extensions at the end.
As the file executes and each sqlloader call completes, the number of data records loaded will be displayed to computer screen.

How to:
You are currently in the step01 directory. Change to the step02 directory:

[Type the following to changes from the step01 directory to the step02 directory.]
%cd ../step02

[*Run the second step*]
%source a2_load_dma_tables *<RETURN>/<EXECUTE>*

[Note: Step07 can be done now. Originally, we obtained the remarks data late so they were done seperately.]

Step03

Scripts executed in this step:
create_arpt_indx.sql
a3_update_dma_tables.sql
Approximate Time*:
1hr.

Description:

Procedure *create_arpt_indx.sql* creates an index on the arpt_pk field in all tables. Script *a3_update_dma_tables.sql* executes a sql script for every table in the database. These scripts update the DMA tables by adding two fields, and filling these fields with the appropriate data. The fields added are the key fields World Area Code (WAC), ins_num_id, and the multi_scty_clsn_cd field. The naming convention for the scripts within the a3_*update_dma_tables.sql* script is '*update_dma'* at the beginning and *.sql* extensions at the end.

How to:
Change to the step03 directory as in step02 above
%cd ../step03 <ENTER>
Start sqlplus
%sqlplus username/password <ENTER>

SQL>@ *do_aprt_indx.sql <RETURN>/<EXECUTE>*

[The index increases Oracle's performance almost 75%. The following script may be run without first creating indexes, but run time would increase from hours to days.]

Next run the *a3_update_dma_tables.sql* script.
SQL>@ *a3_update_dma_tables.sql <RETURN>/<EXECUTE>*

[As the file executes, the update for each table identified in the a3_update_dma_tables file will be displayed to computer screen. This is accomplished on a table by table basis until all tables have been updated.]

Note: As each table's script executes, there is a possibility of duplicate records in each table. If this occurs, the script (for that particular table) will not complete. The next step will help identify the duplicates. DMA is aware of this problem and steps are being taken on their end to rectify this inconvenience. If no duplicates exist, Step04 may be skipped.

Step04
Scripts executed in this step:
show_duplicates_##.sql
*kill_duplicates_*##.sql
Approximate Time*:

2 min run time per script.
~identification time 1 day. Depending on how many tables have duplicate records.

Description:

Script show_duplicates_##.sql searches a table for duplicate arpt_pk values. The ## represents the table number, which corresponds to the table number in the ctl script, as well as the table number on the DMA printout (mentioned in step02). Consistency is important. If duplicates are found the table_name and the arpt_pk value are stored in a table called Hold_Duplicates, which is created in the script show_duplicates_00.sql. Once duplicates are identified, the scripts kill_duplicates_##.sql may be used, however it is recommended that you manually effect the changes to these tables yourself, and keep a record of all changes made.

How to:
[Change to the step04 directory. And start sqlplus as in previous steps.]

SQL>@show_duplicates_00.sql
[*This creates the hold_duplicates table.*]

SQL>@show_duplicates_##.sql
[where ## corresponds to the table which had duplicate arpt_pk values.]

SQL>select * from hold_duplicates order by table_name;
[This will show you the duplicate arpt_pk values in your table.]

Expert Tip:

Stay in SQL.
SQL> select * from table_name where arpt_pk = arptpk_value;

This will list the records with the same arpt_pk. Compare the values in each field. There are three possibilities:
1. Everything will match.
2. Everything will match, with maybe one exception (usually the mod_time, or add_time field). Ensure that this field is not important to the application. *(In all of our cases the field was not important.)*

In either of the above scenarios, delete one duplicated record, and move on. Upon identification of all duplicates contact DMA and inform them of your findings.

3. Pertinent application information differs. If you find an instance when data pertinent to the application differs, contact DMA immediately, for it may take them a day (or more) to obtain the

correct information. If this occurs, it is recommended that you copy the information of all records with the same arpt_pk value to another table. Delete these records from your main table and complete your load. This record will take mere seconds to load alone... at another time.

Upon completion log out of sql, in preparation for the next step.

Step05
Scripts executed in this step:
a4_create_af_tables.sql
Approximate Time*:
~1hr.

Description:

Script *a4_create_af_tables.sql* contains all scripts that create and load the Airfields (AF) tables. Appropriately, there is one script for every table in the database. The appropriate table is first dropped and then created. The 'drop' is a precautionary step used to ensure that a table, which may contain garbage or is otherwise corrupted or inaccurate, is not utilized. Next data from the Defense Mapping Agency's corresponding table is inserted into the Airfields table and the information is committed. Upon completion of this process, all Airfield tables will be loaded with data. The naming convention of the scripts contained within a4_*create_af_tables.sql* is characterized by the words ' *create_af'* at the beginning concatenated with the table_name and *.sql* extensions at the end. Script *a4_create_af_tables.sql* should be executed under Oracle SQL Plus.

How To:
[Change to the step05 directory. And start sqlplus as in previous steps.]

SQL>@*a4_create_af_tables.sql <RETURN>/<EXECUTE >*

NOTE: All Airfield tables are created and loaded in this step EXCEPT the AP_SCTY_CLSN (Security) table.

As the above procedure executes, each table, as it is first dropped and then created, will be displayed to computer screen. Scripts for the following tables are included in the *a4_create_af_tables.sql* file:

aircraft bunkers
airport
apron
arresting system
defueling

fuel dispensing
fuel stock
hangars
hardstand
off base facility storage
oconus airport
oconus runway
oconus taxiway
refueling
revetments
runway
shed
taxiway
warehouse
weather

Step06
Scripts executed in this step:
a1a_create_remark_tables.sql
Approximate Time*:
~20 min.

Description:
The Defense Mapping Agency has remarks in separate tables. Procedure *a1a_create_remark_tables.sql* creates the DMA remarks table structures. This file should be executed under Oracle SQL Plus.

How To:
[Change to the step06 directory. And start sqlplus as in previous steps.]

SQL>*a1a_create_remark_tables.sql<RETURN>/<EXECUTE* >

Note: This could have been done with step01.

Step07
Scripts executed in this step:
a2a_load_dma_remarks
Approximate Time*:
1.5 hr.

Description:
Same as step02 above, however this is remarks data.

How To:
[Same as step02, however execute from step07 directory]

%cd ../step07
%source *a2a_load_dma_remarks <RETURN>/<EXECUTE>*

Step08
Scripts executed in this step:
a3a_update_dma_remarks_tables.sql
Approximate Time*:
~4 hr.

Description:
Procedure *a3a_update_dma_remarks_tables.sql* adds the appropriate keys to the dma_remark_tables. And initially loads the incremental pointer to each remark. This procedure must be executed under Oracle SQL Plus. **Ensure that indexes exist for the DMA remarks tables on the arpt_pk field**. Each update should only take ~ 5min. If duplicates exist follow the procedures in step04 above. The scripts which find duplicate remarks differ slightly, but you will find examples in the step04 directory.

How To:
[Change to the step08 directory. And start sqlplus as in previous steps.]

SQL>@*a3a_update_dma_remarks_tables.sql<RETURN>*

Step09
Scripts executed in this step:
a8_security_remark.sql
Approximate Time*:
~2 hr. ~5min per script

Description:
Script *a8_security_remark.sql* creates the Airfields Security Classification table (AP_SCTY_CLSN) and loads it from the DMA tables created and loaded above. This script must be executed under Oracle SQL Plus.

How To:
[Change to the step09 directory. And start sqlplus as in previous steps.]

SQL>*a8_security_remark.sql <RETURN>/<EXECUTE>*

Step10
Scripts executed in this step:
miscellaneous
Approximate Time*:
variable

Description
This directory is reserved for those scripts which had problems. (ie, missing data files. These are the ones needed in the initial

load. Use these as a guideline.)

Cleanup (step11)

Scripts executed in this step:
cleanup.sql
copyover.sql
describe.sql
describe_unclass.sql
drop_dma.sql
make_unk.sql
table_names.lst
table_descriptions.lst
unclass_table_descriptions.lst

Approximate Time*:
~1.5 hr. ~5min per script

Description:
These scripts are scripts which will be used to cleanup the database of all intermediate tables used to complete this load, as well as miscellaneous scripts which may be needed throughout.

Cleanup
drops all DMA tables.

copyover.sql
makes a copy of all airfields tables of the name ‘hold_’ concatenated with the airfields_table_name.

describe.sql
describes each of the 24 airfields tables.

describe_unclass.sql
describes the unclassified versions of the airfields tables.

drop_dma.sql
drops all DMA tables.

make_unk.sql
sets the multi_scty_clsn_cd field of each airfields table to blanks.

table_names.lst
list of all table names DMA and airfields.